UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/045,337	01/14/2002	Jimmie Earl DeWitt JR.	AUS920010717US1	3856

7590 10/07/2004

Joseph R. Burwell
Law Office of Joseph R. Burwell
P.O. Box 28022
Austin, TX 78755-8022

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 10/07/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/045,337	DEWITT ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 January 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 January 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *14 January 2002*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-37 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: IDS as received on 1/14/2002.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 7, 17, 27, and 34 recite the limitation "the instruction disposition value". There is insufficient antecedent basis for this limitation in the claim because “the instruction disposition value” may refer to either “an instruction disposition value” disclosed in each of the aforementioned claims or it may refer to “an instruction disposition value” disclosed in each of the respective parent claims. The examiner asserts that “an instruction disposition value” in claims 7, 17, 27, and 34 should either be changed to “the instruction disposition value” or, if the

applicant has meant this to mean obtaining one of a plurality of instruction disposition values, then applicant should modify the claims accordingly.

6. Claims 8-10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, applicant's claim 8 claims "...automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction". However, the examiner asserts that if the writing occurs when the instruction's execution is complete, the writing cannot occur while the instruction is executing. For purposes of this examination, the examiner will interpret claim 8 as if it did not include the words "while executing the instruction". Therefore, it becomes more similar to claim 1 (with the additional predicate limitations).

7. Claims 18-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, applicant's claim 18 claims "...means for automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction". However, the examiner asserts that if the writing occurs when the instruction's execution is complete, the writing cannot occur while the instruction is executing. For purposes of this examination, the examiner will interpret claim 18 as if it did not include the words "while executing the instruction". Therefore, it becomes more similar to claim 1 (with the additional predicate limitations).

8. Claims 28-37 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, in claims 28 and 35, applicant claims "A computer program product in a computer-readable medium for use in a processor, the computer program product comprising: means for executing an instruction within the processor..." It is not clear to the examiner how the computer program or the computer-readable medium includes a means for executing an instruction within the processor. More specifically, the computer program itself is executed by the processor, i.e., it does not have means for executing an instruction within the processor. In addition, the computer-readable medium provides instructions for execution instead of including a means for executing (it would be a means for providing).

9. Claims 35-37 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, applicant's claim 35 claims "...means for automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction". However, the examiner asserts that if the writing occurs when the instruction's execution is complete, the writing cannot occur while the instruction is executing. For purposes of this examination, the examiner will interpret claim 35 as if it did not include the words "while executing the instruction". Therefore, it becomes more similar to claim 1 (with the additional predicate limitations).

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. Claim 1, 4-7, 11, 14-17, 21, and 24-27 are rejected under 35 U.S.C. 102(b) as being anticipated by Tran et al., U.S. Patent No. 5,881,278 (herein referred to as Tran).

12. Referring to claim 1, Tran has taught a method for processing an instruction within a processor, the method comprising: executing an instruction within the processor; and in response to completion of the executed instruction, automatically writing by the processor an instruction disposition value to a register or to a memory buffer, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 12, lines 2-5. Note that when a call instruction is committed (via reorder buffer), a memory bit is set.

13. Referring to claim 4, Tran has taught a method as described in claim 1. Tran has further taught determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 11, lines 63-65. Note that the CV flag indicates a call has been detected and that the call information is valid. Clearly, if a call instruction is to be retired, then it must be checked for validity first (that is, the system will not retire invalid information).

14. Referring to claim 5, Tran has taught a method as described in claim 1. Tran has further taught reading a memory buffer pointer register within the process or to obtain a pointer to the memory buffer. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254.

15. Referring to claim 6, Tran has taught a method as described in claim 1. Tran has further taught writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254. For it to be read, it must have been written.

16. Referring to claim 7, Tran has taught a method as described in claim 1. Tran has further taught:

a) reading the register or the memory buffer by tracing software to obtain an instruction disposition value. Clearly, if the CRV is set of some purpose, then it is also read for some purpose; otherwise, it would be a waste of hardware resources.

b) writing the instruction disposition value to persistent storage. Again, see column 12, lines 2-5. Note that to set a CRV bit, it must be written to. Also, this CRV field qualifies as persistent storage because according to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that if the CRV is set, it will remain set until it is changed. Consequently, the CRV is persistent storage.

17. Referring to claim 11, Tran has taught a processor that performs operations specified by instructions fetched from a memory, the processor comprising:

a) means for fetching instructions from memory. See Fig.1 and note the components (202, 204, 206) that are associated with the inherent operation of fetching instructions. Without such fetching, instructions would not be executed and work would not be performed by the processor.

b) means for executing an instruction within the processor. See Fig.1 and note components 212.

c) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 12, lines 2-5. Note that when a call instruction is committed (via reorder buffer), a memory bit is set.

18. Referring to claim 14, Tran has taught a processor as described in claim 11. Tran has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 11, lines 63-65. Note that the CV flag indicates a call has been detected and that the call information is valid. Clearly, if a call instruction is to be retired, then it must be checked for validity first (that is, the system will not retire invalid information).

19. Referring to claim 15, Tran has taught a processor as described in claim 11. Tran has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254.

20. Referring to claim 16, Tran has taught a processor as described in claim 11. Tran has further taught means for writing a memory address for the memory buffer to a memory buffer

pointer register within the processor. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254. For it to be read, it must have been written.

21. Referring to claim 17, Tran has taught a processor as described in claim 1. Tran has further taught:

a) means for reading the register or the memory buffer by tracing software to obtain an instruction disposition value. Clearly, if the CRV is set of some purpose, then it is also read for some purpose; otherwise, it would be a waste of hardware resources.

b) means for writing the instruction disposition value to persistent storage. Again, see column 12, lines 2-5. Note that to set a CRV bit, it must be written to. Also, this CRV field qualifies as persistent storage because according to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that if the CRV is set, it will remain set until it is changed. Consequently, the CRV is persistent storage.

22. Referring to claim 21, Tran has taught a data processing system comprising:

a) means for enabling tracing of a process within the data processing system. See column 91, lines 26-32, and note that tracing may be performed.

b) means for executing an instruction within the processor. See Fig.1, components 212.

c) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were

committed. See column 12, lines 2-5. Note that when a call instruction is committed (via reorder buffer), a memory bit is set.

d) means for storing tracing information. See column 150, lines 56-59 and note that trace flags exist. Consequently, these flags will store trace information.

23. Referring to claim 24, Tran has taught a system as described in claim 21. Tran has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 11, lines 63-65. Note that the CV flag indicates a call has been detected and that the call information is valid. Clearly, if a call instruction is to be retired, then it must be checked for validity first (that is, the system will not retire invalid information).

24. Referring to claim 25, Tran has taught a system as described in claim 21. Tran has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254.

25. Referring to claim 26, Tran has taught a system as described in claim 21. Tran has further taught means for writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254. For it to be read, it must have been written.

26. Referring to claim 27, Tran has taught a system as described in claim 21. Tran has further taught:

a) means for reading the register or the memory buffer by tracing software to obtain an instruction disposition value. Clearly, if the CRV is set of some purpose, then it is also read for some purpose; otherwise, it would be a waste of hardware resources.

b) means for writing the instruction disposition value to persistent storage. Again, see column 12, lines 2-5. Note that to set a CRV bit, it must be written to. Also, this CRV field qualifies as persistent storage because according to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that if the CRV is set, it will remain set until it is changed. Consequently, the CRV is persistent storage.

27. Claims 1-8, 10-18, and 20-27 are rejected under 35 U.S.C. 102(e) as being anticipated by Cofler et al., U.S. Patent No. 6,754,856 (herein referred to as Cofler).

28. Referring to claim 1, Cofler has taught a method for processing an instruction within a processor, the method comprising: executing an instruction within the processor; and in response to completion of the executed instruction, automatically writing by the processor an instruction disposition value to a register or to a memory buffer, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 2, lines 28-32. Also, see Fig.6 and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33, an instruction disposition value us written to commit-FIFO 121.

29. Referring to claim 2, Cofler has taught a method as described in claim 1. Cofler has further taught that the instruction disposition value is correlated with a predicate value for the executed instruction. See column 7, lines 27-31 and note that the instruction disposition value written to the buffer (FIFO) corresponds to the predicate (guard) true/false value. Therefore, they are correlated. For instance, when the guard is resolved as true, the disposition value will always be X, whereas when the guard is resolved as false, the disposition value will always be Y.

30. Referring to claim 3, Cofler has taught a method as described in claim 2. Cofler has further taught reading the predicate value from a predicate register that was used for a predication operation while the instruction was executing. See Fig.2, component 44. Note that predicate (guard) values are read from predicate (guard) registers.

31. Referring to claim 4, Cofler has taught a method as described in claim 1. Cofler has further taught determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 7, lines 27-31, and note that before the disposition value (commit signal) may be written to FIFO 121 (of Fig.6), the guard must first be resolved as true or false. This guard acts as an enable flag because when its value is known, it enables the commit signal to be written. When the guard is unknown (not resolved), the commit signal must wait to be written to the FIFO.

32. Referring to claim 5, Cofler has taught a method as described in claim 1. Cofler has further taught reading a memory buffer pointer register within the process or to obtain a pointer to the memory buffer. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are

removed. See the attached definition. Consequently, in order to access the head and tail, it is inherent that head and tail pointers must be read.

33. Referring to claim 6, Cofler has taught a method as described in claim 1. Cofler has further taught writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to track the head and tail, it is inherent that head and tail pointers must be written to memory.

34. Referring to claim 7, Cofler has taught a method as described in claim 1. Cofler has further taught:

a) reading the register or the memory buffer by tracing software to obtain an instruction disposition value. See column 5, lines 35-49. Note that the trace software must determine which instructions have completed or not completed. This is done by reading the stored commit signals because they indicate if an instruction is committed.

b) writing the instruction disposition value to persistent storage. The instruction disposition values are written to persistent storage 121 in Fig.6. According to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that a queue has the ability to hold information indefinitely. Consequently, it is persistent storage.

35. Referring to claim 8, Cofler has taught a method for processing an instruction within a processor, wherein the processor has at least one predicate register (see Fig.2, component 44), the method comprising:

a) executing an instruction within the processor. See Fig.2, components 40, 41, 50, and 51.

b) if the instruction is controlled by a predicate register, automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction. See column 2, lines 28-32, and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33 (Fig.6), the predicate register value us written to commit-FIFO 121 (Fig.6). For instance, if the predicate were resolved as "true" then a value of 1 would be written to the FIFO (assuming false and 0 are synonymous). However, if the predicate were resolved as "false," then a value of 0 would be written to the FIFO (assuming false and 0 are synonymous).

36. Referring to claim 10, Cofler has taught a method as described in claim 8. Cofler has further taught storing a series of values in the register or the memory buffer for a series of instructions. See Fig.6 and note that the FIFO has multiple entries for holding multiple values.

37. Referring to claim 11, Cofler has taught a processor that performs operations specified by instructions fetched from a memory, the processor comprising:

a) means for fetching instructions from memory. See Fig.1 and note the components (30, 31, 80, and 32) that are associated with the inherent operation of fetching instructions. Without such fetching, instructions would not be executed and work would not be performed by the processor.

b) means for executing an instruction within the processor. See Fig.1 and note components 40, 41, 50, and 51.

c) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 2, lines 28-32. Also, see Fig.6 and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33, an instruction disposition value us written to commit-FIFO 121.

38. Referring to claim 12, Cofler has taught a processor as described in claim 11. Cofler has further taught that the instruction disposition value is correlated with a predicate value for the executed instruction. See column 7, lines 27-31 and note that the instruction disposition value written to the buffer (FIFO) corresponds to the predicate (guard) true/false value. Therefore, they are correlated. For instance, when the guard is resolved as true, the disposition value will always be X, whereas when the guard is resolved as false, the disposition value will always be Y.

39. Referring to claim 13, Cofler has taught a processor as described in claim 12. Cofler has further taught reading the predicate value from a predicate register that was used for a predication operation while the instruction was executing. See Fig.2, component 44. Note that predicate (guard) values are read from predicate (guard) registers.

40. Referring to claim 14, Cofler has taught a processor as described in claim 11. Cofler has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 7, lines 27-31, and note that before the disposition value (commit signal) may be written to FIFO 121 (of Fig.6), the guard must first be resolved as true or false. This guard acts as an enable flag because when its value is known, it

enables the commit signal to be written. When the guard is unknown (not resolved), the commit signal must wait to be written to the FIFO.

41. Referring to claim 15, Cofler has taught a processor as described in claim 11. Cofler has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to access the head and tail, it is inherent that head and tail pointers must be read.

42. Referring to claim 16, Cofler has taught a processor as described in claim 11. Cofler has further taught means for writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to track the head and tail, it is inherent that head and tail pointers must be written to memory.

43. Referring to claim 17, Cofler has taught a processor as described in claim 1. Cofler has further taught:

a) means for reading the register or the memory buffer by tracing software to obtain an instruction disposition value. See column 5, lines 35-49. Note that the trace software must determine which instructions have completed or not completed. This is done, by reading the stored commit signals because they indicate if an instruction is committed.

b) means for writing the instruction disposition value to persistent storage. The instruction disposition values are written to persistent storage 121 in Fig.6. According to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that a queue has the ability to hold information indefinitely. Consequently, it is persistent storage.

44. Referring to claim 18, Cofler has taught a processor that performs operations specified by instructions fetched from a memory, the processor comprising:

a) means for executing an instruction within the processor. See Fig.2, components 40, 41, 50, and 51.

b) means for automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction if the instruction is controlled by a predicate register. See column 2, lines 28-32, and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33 (Fig.6), the predicate register value us written to commit-FIFO 121 (Fig.6). For instance, if the predicate were resolved as "true" then a value of 1 would be written to the FIFO (assuming false and 0 are synonymous). However, if the predicate were resolved as "false," then a value of 0 would be written to the FIFO (assuming false and 0 are synonymous).

45. Referring to claim 20, Cofler has taught a processor as described in claim 18. Cofler has further taught means for storing a series of values in the register or the memory buffer for a

series of instructions. See Fig.6 and note that the FIFO has multiple entries for holding multiple values.

46. Referring to claim 21, Cofler has taught a data processing system comprising:

a) means for enabling tracing of a process within the data processing system. See column 5, lines 27-49, and note that tracing may be performed.

b) means for executing an instruction within the processor. See Fig.1, components 40, 41, 50, and 51.

c) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 2, lines 28-32. Also, see Fig.6 and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33, an instruction disposition value us written to commit-FIFO 121.

d) means for storing tracing information. Clearly, if a trace is performed, then trace information is stored.

47. Referring to claim 22, Cofler has taught a system as described in claim 21. Cofler has further taught that the instruction disposition value is correlated with a predicate value for the executed instruction. See column 7, lines 27-31 and note that the instruction disposition value written to the buffer (FIFO) corresponds to the predicate (guard) true/false value. Therefore, they are correlated. For instance, when the guard is resolved as true, the disposition value will always be X, whereas when the guard is resolved as false, the disposition value will always be Y.

48. Referring to claim 23, Cofler has taught a system as described in claim 22. Cofler has further taught means for reading the predicate value from a predicate register that was used for a predication operation while the instruction was executing. See Fig.2, component 44. Note that predicate (guard) values are read from predicate (guard) registers.

49. Referring to claim 24, Cofler has taught a system as described in claim 21. Cofler has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 7, lines 27-31, and note that before the disposition value (commit signal) may be written to FIFO 121 (of Fig.6), the guard must first be resolved as true or false. This guard acts as an enable flag because when its value is known, it enables the commit signal to be written. When the guard is unknown (not resolved), the commit signal must wait to be written to the FIFO.

50. Referring to claim 25, Cofler has taught a system as described in claim 21. Cofler has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to access the head and tail, it is inherent that head and tail pointers must be read.

51. Referring to claim 26, Cofler has taught a system as described in claim 21. Cofler has further taught means for writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a

tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to track the head and tail, it is inherent that head and tail pointers must be written to memory.

52. Referring to claim 27, Cofler has taught a system as described in claim 21. Cofler has further taught:

a) means for reading the register or the memory buffer by tracing software to obtain an instruction disposition value. See column 5, lines 35-49. Note that the trace software must determine which instructions have completed or not completed. This is done by reading the stored commit signals because they indicate if an instruction is committed.

b) means for writing the instruction disposition value to persistent storage. The instruction disposition values are written to persistent storage 121 in Fig.6. According to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that a queue has the ability to hold information indefinitely. Consequently, it is persistent storage.

Claim Rejections - 35 USC § 103

53. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

54. Claims 28 and 31-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tran, as applied above, in view of Tanenbaum, "Structured Computer Organization, 2nd Edition," 1984 (herein referred to as Tanenbaum).

55. Referring to claim 28, Tran has taught a system (hardware) comprising:

a) means for executing an instruction within the processor. See Fig.1, components 212.

b) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 12, lines 2-5. Note that when a call instruction is committed (via reorder buffer), a memory bit is set.

c) Tran has not taught that a computer program product in a computer readable medium for use in a processor comprises the aforementioned means. However, Tanenbaum has taught that hardware and software are logically equivalent. See page 11. Therefore, a person of ordinary skill in the art would've recognized that the function performed by the hardware could be implemented in software and vice-versa. As Tanenbaum has further suggested, the choice between hardware and software implementations is based on the designer's needs as well as cost, speed, reliability, and frequency. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Tran such that a computer program product comprises means instead of the hardware.

56. Referring to claim 31, Tran in view of Tanenbaum has taught a computer program product as described in claim 28. Tran has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column

11, lines 63-65. Note that the CV flag indicates a call has been detected and that the call information is valid. Clearly, if a call instruction is to be retired, then it must be checked for validity first (that is, the system will not retire invalid information).

57. Referring to claim 32, Tran in view of Tanenbaum has taught a computer program product as described in claim 28. Tran has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254.

58. Referring to claim 33, Tran in view of Tanenbaum has taught a computer program product as described in claim 28. Tran has further taught means for writing a memory address for the memory buffer to a memory buffer pointer register within the processor. See Fig.2 and note that a buffer pointer is read from unit 254 and sent along pointer bus 254. For it to be read, it must have been written.

59. Referring to claim 34, Tran in view of Tanenbaum has taught a computer program product as described in claim 28. Tran has further taught:

a) means for reading the register or the memory buffer to obtain an instruction disposition value. Clearly, if the CRV is set of some purpose, then it is also read for some purpose; otherwise, it would be a waste of hardware resources.

b) means for writing the instruction disposition value to persistent storage. Again, see column 12, lines 2-5. Note that to set a CRV bit, it must be written to. Also, this CRV field qualifies as persistent storage because according to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that if

the CRV is set, it will remain set until it is changed. Consequently, the CRV is persistent storage.

60. Claims 9 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cofler, as applied above, and further in view of Intel, "IA-64 Application Developer's Architecture Guide," May 1999 (herein referred to as Intel).

61. Referring to claim 9, Cofler has taught a method as described in claim 8. Cofler has not explicitly taught that if the instruction is not controlled by a predicate register, automatically writing by the processor to a register or to a memory buffer a value that indicates that the instruction was fully executed. However, Intel has taught that some instructions within an instruction set may be predicated (see Intel, pages 7-7 and 7-149, for instance - note the use of predicate qp) and some instructions may not be predicated (see Intel, pages 7-5 and 7-18, for instance - note the lack of predicate qp). It would have been obvious to one of ordinary skill in the art at the time of the invention to have some predicated and some non-predicated instructions in Cofler, because as Intel has taught some instructions cannot or should not be predicated, and predicating them would result in a waste of time and resources. By having these non-predicated instructions, more functionality would be added to the system of Cofler. In addition, Cofler has taught the desire to have a trace of all instructions which are fetched and committed (column 5, lines 35-49), and therefore, if some instructions are not predicated (as taught by Intel), then it would have been further obvious to write a value to storage saying whether the non-predicated instruction is fully executed. This would be the only way for a trace program to determine if a non-predicated instruction has completed.

62. Referring to claim 19, Cofler has taught a processor as described in claim 18. Cofler has not explicitly taught means for automatically writing by the processor to a register or to a memory buffer a value that indicates that the instruction was fully executed if the instruction is not controlled by a predicate register. However, Intel has taught that some instructions within an instruction set may be predicated (see Intel, pages 7-7 and 7-149, for instance - note the use of predicate qp) and some instructions may not be predicated (see Intel, pages 7-5 and 7-18, for instance - note the lack of predicate qp). It would have been obvious to one of ordinary skill in the art at the time of the invention to have some predicated and some non-predicated instructions in Cofler, because as Intel has taught some instructions cannot or should not be predicated, and predicating them would result in a waste of time and resources. By having these non-predicated instructions, more functionality would be added to the system of Cofler. In addition, Cofler has taught the desire to have a trace of all instructions which are fetched and committed (column 5, lines 35-49), and therefore, if some instructions are not predicated (as taught by Intel), then it would have been further obvious to write a value to storage saying whether the non-predicated instruction is fully executed. This would be the only way for a trace program to determine if a non-predicated instruction has completed.

63. Claims 28-35 and 37 rejected under 35 U.S.C. 103(a) as being unpatentable over Cofler, as applied above, in view of Tanenbaum, as applied above.

64. Referring to claim 28, Cofler has taught a system (hardware) comprising:

a) means for executing an instruction within the processor. See Fig.1, components 40, 41, 50, and 51.

b) means for automatically writing by the processor an instruction disposition value to a register or to a memory buffer in response to completion of the executed instruction, wherein the instruction disposition value indicates whether results from the executed instruction were committed. See column 2, lines 28-32. Also, see Fig.6 and column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33, an instruction disposition value us written to commit-FIFO 121.

c) Cofler has not taught that a computer program product in a computer readable medium for use in a processor comprises the aforementioned means. However, Tanenbaum has taught that hardware and software are logically equivalent. See page 11. Therefore, a person of ordinary skill in the art would've recognized that the function performed by the hardware could be implemented in software and vice-versa. As Tanenbaum has further suggested, the choice between hardware and software implementations is based on the designer's needs as well as cost, speed, reliability, and frequency. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cofler such that a computer program product comprises means instead of the hardware.

65. Referring to claim 29, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. Cofler has further taught that the instruction disposition value is correlated with a predicate value for the executed instruction. See column 7, lines 27-31 and note that the instruction disposition value written to the buffer (FIFO) corresponds to the predicate (guard) true/false value. Therefore, they are correlated. For instance, when the guard is resolved as true, the disposition value will always be X, whereas when the guard is resolved as false, the disposition value will always be Y.

66. Referring to claim 30, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. Cofler has further taught reading the predicate value from a predicate register that was used for a predication operation while the instruction was executing. See Fig.2, component 44. Note that predicate (guard) values are read from predicate (guard) registers.

67. Referring to claim 31, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. Cofler has further taught means for determining whether or not an enable flag was previously set prior to writing the instruction disposition value. See column 7, lines 27-31, and note that before the disposition value (commit signal) may be written to FIFO 121 (of Fig.6), the guard must first be resolved as true or false. This guard acts as an enable flag because when its value is known, it enables the commit signal to be written. When the guard is unknown (not resolved), the commit signal must wait to be written to the FIFO.

68. Referring to claim 32, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. Cofler has further taught means for reading a memory buffer pointer register within the processor to obtain a pointer to the memory buffer. See column 8, lines 30-34, and note that the FIFOs are ordered queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to access the head and tail, it is inherent that head and tail pointers must be read.

69. Referring to claim 33, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. See column 8, lines 30-34, and note that the FIFOs are ordered

queues. A queue, according to The Free On-Line Dictionary Of Computing, 1995, has a tail, to which objects (disposition values) are added, and a head, from which objects (disposition values) are removed. See the attached definition. Consequently, in order to track the head and tail, it is inherent that head and tail pointers must be written to memory.

70. Referring to claim 34, Cofler in view of Tanenbaum has taught a computer program product as described in claim 28. Cofler has further taught:

a) means for reading the register or the memory buffer to obtain an instruction disposition value. See column 5, lines 35-49. Note that the trace software must determine which instructions have completed or not completed. This is done by reading the stored commit signals because they indicate if an instruction is committed.

b) means for writing the instruction disposition value to persistent storage. The instruction disposition values are written to persistent storage 121 in Fig.6. According to The American Heritage Dictionary of the English Language, 3rd Edition, 1992, "persistent" is defined as "existing or remaining in the same state for an indefinitely long time". See the attached definition. Therefore, it should be realized that a queue has the ability to hold information indefinitely. Consequently, it is persistent storage.

71. Referring to claim 35, Cofler has taught a system (hardware) comprising:

a) means for executing an instruction within the processor. See Fig.2, components 40, 41, 50, and 51.

b) means for automatically writing by the processor to a register or to a memory buffer a value of the predicate register while executing the instruction in response to completion of the executed instruction if the instruction is controlled by a predicate register. See column 2, lines 28-32, and

column 7, lines 23-31. Note that, in the given example, when the executed instruction completes the e2 stage of the pipeline 33 (Fig.6), the predicate register value us written to commit-FIFO 121 (Fig.6). For instance, if the predicate were resolved as "true" then a value of 1 would be written to the FIFO (assuming false and 0 are synonymous). However, if the predicate were resolved as "false," then a value of 0 would be written to the FIFO (assuming false and 0 are synonymous).

c) Cofler has not taught that a computer program product in a computer readable medium for use in a processor comprises the aforementioned means. However, Tanenbaum has taught that hardware and software are logically equivalent. See page 11. Therefore, a person of ordinary skill in the art would've recognized that the function performed by the hardware could be implemented in software and vice-versa. As Tanenbaum has further suggested, the choice between hardware and software implementations is based on the designer's needs as well as cost, speed, reliability, and frequency. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Cofler such that a computer program product comprises means instead of the hardware.

72. Referring to claim 37, Cofler in view of Tanenbaum has taught a computer program product as described in claim 35. Cofler has further taught means for storing a series of values in the register or the memory buffer for a series of instructions. See Fig.6 and note that the FIFO has multiple entries for holding multiple values.

73. Claim 36 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cofler in view of Tanenbaum, as applied above, and further in view of Intel, as applied above.

74. Referring to claim 36, Cofler in view of Tanenbaum has taught a computer program product as described in claim 35. Cofler in view of Tanenbaum has not explicitly taught means for automatically writing by the processor to a register or to a memory buffer a value that indicates that the instruction was fully executed if the instruction is not controlled by a predicate register. However, Intel has taught that some instructions within an instruction set may be predicated (see Intel, pages 7-7 and 7-149, for instance - note the use of predicate qp) and some instructions may not be predicated (see Intel, pages 7-5 and 7-18, for instance - note the lack of predicate qp). It would have been obvious to one of ordinary skill in the art at the time of the invention to have some predicated and some non-predicated instructions in Cofler, because as Intel has taught some instructions cannot or should not be predicated, and predicating them would result in a waste of time and resources. By having these non-predicated instructions, more functionality would be added to the system of Cofler. In addition, Cofler has taught the desire to have a trace of all instructions which are fetched and committed (column 5, lines 35-49), and therefore, if some instructions are not predicated (as taught by Intel), then it would have been further obvious to write a value to storage saying whether the non-predicated instruction is fully executed. This would be the only way for a trace program to determine if a non-predicated instruction has completed.

Conclusion

75. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the

references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Mann, U.S. Patent No. 6,314,530, has taught a processor having a trace access instruction to access on-chip trace memory.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
September 23, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100